# "Method and circuit for data encryption/decryption"

* * *

## Field of the invention

The invention relates to encryption/decryption techniques and more specifically refers to Advanced Encryption Standard (AES) cryptosystems based e.g. on the so-called Rijndael algorithm.

The Rijndael algorithm is a block cipher algorithm operating on blocks of data. The algorithm reads an entire block of data, processes the block and then outputs the encrypted data. The Rijndael algorithm needs a key, which is another block of data. The proposed AES standard will include only 128-bit as standard length for plaintext blocks and 128, 192 and 256-bit as standard lengths for the key material.

## Description of the prior art

For a general review of the Rijndael/AES algorithms reference may be made to the following documents/websites:

J. Daemen, V. Rijmen, "AES Proposal: Rijndael" www.nist.gov/aes;

J. Daemen, V. Rijmen, "The Block Cipher Rijndael" Smart Card Research and Applications, LNCS 1820, J.-J. Quisquater and B. Schneier, Eds., Springer-Verlag, 2000, pp. 288-296;

J. Daemen and V. Rijmen, "Rijndael, the advanced encryption standard", Dr. Dobb's Journal, Vol.~26, No. ~3, March 2001, pp. 137-139;

V. Rijmen, "Efficient Implementation of the Rijndael S-box" http://www.eas.kuleuven.ac.be/~rijmen/rijndael/;

J. Gladman "A specification for Rijndael, the AES Algorithm" March 2001 http://fp.gladman.plus.com/;

M. Akkar, C. Giraud "An implementation of DES and AES, secure against some attacks" - Proceedings of CHES 2001, pp. 315-325;

M. McLoone, J. V. McCanny "High performance single-chip FPGA Rijndael algorithm implementations" - Proceedings of CHES 2001, pp. 68-80;

V. Fischer, M. Drutarovsky "Two methods of Rijndael implementation in reconfigurable Hardware" Proceedings of CHES 2001, pp. 81-96;

H. Kuo and I. Verbauwhede "Architectural optimization for a 3Gbits/sec VLSI Implementation of the AES Rijndael algorithm", Proceedings of CHES 2001, pp. 53-67;

Rudra, P. K. Dubey, C. S. Jutla, V. Kumar, J. R. Rao, and P. Rohatgi "Efficient Rijndael encryption implementation with composite field arithmetic" Proceedings of CHES 2001, pp.175-188;

A. Dandalis, V.K. Prasanna, J.P.D. Rolim "An adaptive cryptographic engine for IPSec architecutures" Field-Programmable Custom Computing Machines, 2000 IEEE Symposium on 2000, pp. 132-141;

"Advanced Encryption Standard (AES)" www.nist.gov/aes;

National Institute of Standard and Technology www.nist.gov/aes

Rijndael Home Page's www.esat.kuleuven.ac.be/rijmen/rijndael/

Gladman Home Page http://fp.gladman.plus.com/

The encryption process based on the Rijndael algorithm follows the general layout shown in figure 1 of the enclosed drawings.

Unencrypted data are subject to a sequence of "rounds" R1, R2, ..., R9, R10. Each round in turn provides for the application of a respective round key (i.e. round key 1, round key 2, ...) generated according to a key scheduling process KS.

Each generic round Ri develops along the lines shown in figure 2 and is essentially based on a first processing step currently referred to as the S-box step or function. This generates a matrix array which is subjected to a row shifting process followed by column mixing.

The respective key scheduled for round Ri is then added to produce the output of the round. The output of the final round (designated round 10 in figure 1) corresponds to the encrypted data.

More specifically, the first and last rounds are at least marginally different from the other rounds: the first round is in fact comprised of key addition only, while the last round does not provide for mix column transformation.

The decryption algorithm of AES is very similar to the encryption process just described. The decryption process is essentially based on a sequence of steps reproducing in a complementary manner the sequence of steps of the encryption process, wherein each transformation is replaced by the respective inverse transformation.

All of the foregoing corresponds to basic principles and criteria well known to those of skill in the art (see, for instance, the references cited in the introductory portion of this description), thus making it unnecessary to provide a more detailed description herein. This applies more to the point to the steps/functions designated "S-box" and "Add Key" in figure 2.

Figure 3 is a schematic representation of a round in matrix form.

Apart from the add round key, sub byte and shift row operations, the application of a single round can essentially be described as the application to an array of input data ID of a matrix M to generate a corresponding array of output data OD. Data ID and OD are in typical 32-bit format partitioned in four 8-bit words (bytes).

In current implementations of the Rijndael/AES algorithm, matrix M is thus a matrix including 4x4 = 16 elements $s_0$, ..., $s_{15}$ is corresponding to a byte.

The block diagram of figure 4 shows a typical embodiment of an encryption system implementing the Rijndael/AES algorithm according to the traditional approach followed so far.

The system shown in figure 4, designated 10 overall, is intended to generate encrypted data starting from unencrypted data UD. Both unencrypted and encrypted data UD and ED are arranged in a 32-bit word format.

In the diagram of figure 4, reference numeral 12 designates a demux unit which distributes the input unencrypted data stream UD over four different paths leading to respective adder modules 14a, 14b, 14c and 14d where the first key addition is performed.

Reference numerals 24a, 24b, 24c and 24d designates respective sets of byte register wherein the 32-bit words subjected to the first key addition are distributed over four byte registers to be subsequently fed to respective sets of modules 34a, 34b, 34c and 34d where the S-box processing takes place.

Reference 16 designates a module which implements the shift row operation. Data blocks resulting from row shifting are fed to respective mix column modules 18a, 18b, 18c and 18d.

These latter modules are intended to be bypassed during the last round. In fact the structure shown permits the first round to be calculated immediately. Iterative calculation is then carried out for the following rounds. As indicated, the last round does not provide for the mix column step, whereby lines are shown enabling such a step to be bypassed during the last round.

The data output from modules 18a, 18b, 18c and 18d - which are arranged over four parallel 8-bit words - are then fed to respective key addition modules 20a, 20b, 20c and 20d where the key addition operation is performed. After being subjected to key addition in modules 20a, 20b, 20c and 20d data are loaded into final registers 22a to 22d from which the encrypted code words are fed to a multiplexer unit 26 to generate the encrypted data stream ED.

All of the foregoing again corresponds to principles and criteria which are known to those of skilled in the art.

The main disadvantage of the prior art solutions exemplified by the arrangement shown in figure 4 lies in the complex circuitry required to implement the encryption/decryption mechanism. Such a disadvantage is particularly felt to those envisaged applications of cryptosystems adapted for use in embedded systems such as e.g. smartcards and the like.

One main object of the present invention is thus to provide an improved form of implementing the Rijndael/AES algorithm making it possible to expand the field of use of such algorithm in cryptosystems.

According to the present invention, this object, as well as additional objects which will become apparent from the following detailed description of a preferred embodiment of the invention, are achieved by means of a method and system having the features set forth in the annexed claims.

The arrangement of the invention can in fact be regarded as embodying a novel encryption method, which however can be rendered compatible with existing standards through initial and final transposition steps.

Detailed description of the drawings

The invention will now be described, by way of non limiting example, by referring to the enclosed drawings, wherein:

- Figures 1 to 4, exemplary of prior art approaches for implementing the Rijndael/AES algorithm have been already described in the foregoing,

- Figure 5 is intended to highlight, by direct comparison to figure 3, the basic underlying mechanism of the present invention, and

- Figure 6 shows how the system shown in the block diagram of figure 4 is modified and simplified by resorting to the present invention.

Detailed description of a preferred embodiment of the invention

In order to better understand the basic underlying principle of the invention, it must be recalled that Rijndael is a secret key cryptographic algorithm working in block cipher mode. This means that it operates on blocks of data and not on single bits or bytes. The algorithm reads an entire block, processes it and then output the encrypted block. The encryption operates in a complementary way to re-obtain plaintext starting from encrypted data.

To operate properly, the Rijndael algorithm needs a key, which is another block of data.

The initial specification for this algorithm includes 128-bit, 192-bit and 256-bit as possible lengths for the plaintext blocks and for the key material. The prospected AES standard will expectedly include only 128-bit as standard length for plaintext blocks and 128, 192 and 256-bit as standard length for the key material.

The following description will therefore deal - by way of example only - with 128-bit blocks, as this adheres to the presently prospected standard.

The input, output and cipher key bit sequences are processed as arrays of bytes formed by dividing these sequences into groups of 8 contiguous bits (bytes). Internally, the operations of the AES algorithm are performed on a two dimensional array of bytes called the state.

Specifically, by referring again to figure 3, matrix ID represents the input bytes, matrix M represents the state bytes, and OD designates the output bytes. The state consists of four rows of bytes, each row containing 4 bytes, thus making the state a 4x4 matrix.

The four bytes in each column of the state array M form 32-bit words, hence the state can also be interpreted as a one-dimensional array of 32-bit words (columns), where the column number provides an index into this array.

As shown in connection with figure 2, the Rijndael cipher algorithm operates in rounds. Each round is a fixed set of transformations that are applied to the state.

The number of these rounds is chosen as a function of the key length. In the case of the three examples referred to in the foregoing, three possible key sizes of 128-bit, 196 and 256 bits can be considered. Depending on these sizes, 10 rounds (as shown in figure

1), 12 rounds or 14 rounds are to be computed, respectively.

The present invention is based on the unexpected recognition that using for the internal state array a transposed arrangement (that is, using -- in the place of matrix M -- matrix M′ where the rows have been exchanged for the columns and vice-versa) leads to a surprising speed-up and simplification of the encryption/decryption process.

According to the prior art, an operation is applied to the columns, for instance column $S_0$ $S_1$ $S_2$ $S_3$ of matrix M1.

When the state is transposed, the column becomes $S_0$ $S_4$ $S_8$ $S_{12}$.

This concept may be better understood by referring to the example which follows of a transformation carried out on a non-transposed state.

$$\begin{bmatrix} S_{0,c} \\ S_{1,c} \\ S_{2,c} \\ S_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} S_{0,c} \\ S_{1,c} \\ S_{2,c} \\ S_{3,c} \end{bmatrix}$$

where c is the column index which can be equal to 0, 1, 2, and 3.

If a new, transposed form is used, the main transformation for the new mix column becomes

$$\begin{aligned} y_0 &= (\{02\}\bullet x_0)+(\{03\}\bullet x_1)+x_2+x_3 \\ y_1 &= x_0+(\{02\}\bullet x_1)+(\{03\}\bullet x_2)+x_3 \\ y_2 &= x_0+x_1+(\{02\}\bullet x_2)+(\{03\}\bullet x_3) \\ y_3 &= (\{03\}\bullet x_0)+x_1+x_2+(\{02\}\bullet x_3) \end{aligned}$$

Transposed Form $\quad x_i = \lfloor S_{0,i}\ S_{1,i}\ S_{2,i}\ S_{3,i} \rfloor$

where $x_i$, $0 \le i \le 3$ are the words of the transposed state, and $y_i$, $0 \le i \le 3$ are the words of the transposed state after mix column transformation.

In the foregoing, operator • means a multiplication in a Galois field applied to each of the four 8-bit terms comprising the 32-bit words being processed (i.e. $\{02\}\bullet x_0$ means $\{02\}\bullet S_{0,0}$ $\{02\}\bullet S_{1,0}$ $\{02\}\bullet S_{2,0}$ $\{02\}\bullet S_{3,0}$) while the operator + is a sum in Galois Fields, a logic XOR between two 32-bit words.

Such a transposition requires a redefinition of must of the operations performed in a round of the algorithm, and also of the key schedule. Therefore, also the round keys must be transposed before being applied to a round providing for the use of a transposed state.

A trivial solution for that purpose is simply to apply the original key schedule unchanged and then add code to transpose every created round key. In that way, a large overhead would be introduced.

For that reason, the preferred embodiment of the invention provides for the key schedule being applied directly in the transposed manner.

This means that the internal behaviour of the system is modified, and simplified, the only requirement to obtain compatibility with the standard being that the state must be re-transposed before being output.

The block diagram of figure 6 shows how the prior art arrangement shown in figure 4 is simplified and rendered faster by resorting to the invention.

In figure 6 parts and components which are identical or equivalent to those already described in connection with figure 4 have been indicated with the same reference numerals.

Essentially, the solution of the invention has a basic impact on the shift row block 16 and the mix column blocks 18a, 18b, 18c and 18d of figure 4.

In the solution of the invention, four shift column modules 16a, 16b, 16c and 16d - each acting on a respective flow from one of the S-box modules 34a, 34b, 34c and 34d - are substituted for shift row module 16.

By referring to the two tables reproduced in the foregoing, it will become apparent that in the solution of the invention generation of each of the components $y_0$ $y_1$ $y_2$ $y_3$ essentially derives from a linear combination of words $x_0$ $x_1$ $x_2$ $x_3$. This makes it possible to implement the respective transformation simply by means of adder modules (and shift registers).

In the block diagram of figure 6 a single mix column module 18 is provided jointly operating on all of the sixteen 8-bit words output from shift column modules 16a, 16b, 16c, 16d is substituted for mix column modules 18a, 18b, 18c and 18d of the prior art arrangement of figure 4.

Experimentation carried out by the applicants demonstrates that the invention significantly increases the speed of implementing the Rijndael algorithm, even if the overhead due to the initial and final transpositions of the state array is taken into account.

Direct comparison of the solution of the invention with the so-called Gladman's implementation (reportedly the fastest soft implementation of the Rijndael algorithm currently available) shows that the invention leads to improvements in terms of encryption and decryption speeds of 46% and 33%, respectively, for a 128-bit key size.

Improvements demonstrated in encryption and decryption speeds with a 192-bit key size are 39% and 25%, respectively.

Finally, improvements in encryption and decryption speed of 45% and 32%, respectively were demonstrated for a 256-bit key size.

It will be appreciated that advantages in terms of latency are primarily felt at the level of software implementation, while the main advantage at the hardware level lies (even with identical performance in terms of latency) in the smaller amount of functional units required. This leads to simpler and less expensive systems, which is a particularly relevant factor in the case of decryption systems.

The solution of transposing the state matrix can be applied to all cases contemplated by the Rijndael algorithm, advantages being significant especially for 128 and 256 bit words. As indicated, if no initial and final transpositions to ensure compatibility with the existing standards are effected, a thoroughly novel cryptographic systems is obtained.

The present invention has been described with reference to the preferred embodiments. However, the present invention is not limited to those embodiments. Various changes and modifications may be made within the spirit and scope of the amended claims.